BSIR 88-3836

THE HIGH LEVEL MACHINE-TOOL CONTROL SYSTEM

ugust 10, 1988

By:
Douglas Fishman

U.S. DEPARTMENT OF COMMERCE National Bureau of Standards Gaithersburg, Maryland

THE HIGH LEVEL MACHINE-TOOL CONTROL SYSTEM

Douglas Fishman

10 August 1988

TABLE OF CONTENTS

TABLE OF CONTENTS (continued)

LIST OF FIGURES

II. OVERVIEW

1. DESCRIPTION OF THE MACHINE-TOOL CONTROLLER

The High Level Machine-tool Controller (HLMC) in the Horizontal Workstation (HWS) is the controller for the machining center equipment component of the workstation (Figure 1). The HLMC receives commands from the Horizontal Workstation Controller (HWSC) to perform machining center tasks and reports status back to the HWSC indicating how well it is performing the current task.

The machining center tasks that the HLMC performs are: positioning a pallet for robotic part loading and unloading, machining a part, and positioning the tool storage drum and actuating the tool holding fingers for robotic tool loading and unloading.

The HLMC has serial and parallel interfaces to the White-Sundstrand horizontal-axis machining center's computer. The HLMC decomposes commands from the HWSC into the tasks that a manual operator would perform at the White-Sundstrand's control panel. It sends commands to perform those tasks to the machining center over the serial and parallel communications links. The HLMC also uses these interfaces to extract status information from the White-Sundstrand and translate it into a form that is appropriate for the HWSC.

The HLMC also has an interface with the Integrated Manufacturing Data Administration System (IMDAS), a remote data base. When connected to the IMDAS, the HLMC can retrieve, in real time, lists of part machining programs (resource lists) required to produce an HWS part, as well as the part programs themselves. As an off-line function, the HLMC can upload resource lists and part programs to the remote data base.

The HLMC uses the AMRF network to communicate with the HWSC and the IMDAS.

2. WHAT THE HORIZONTAL WORKSTATION DOES

The Horizontal Workstation in the AMRF was designed to take commands from a higher level Cell Controller to configure its workstation for part machining and to machine parts. The equipment in the workstation consists of a horizontal-axis machining center, automated part fixturing components on the pallets of the machining center, a robot, and an automated storage and retrieval system for material buffering. Each of

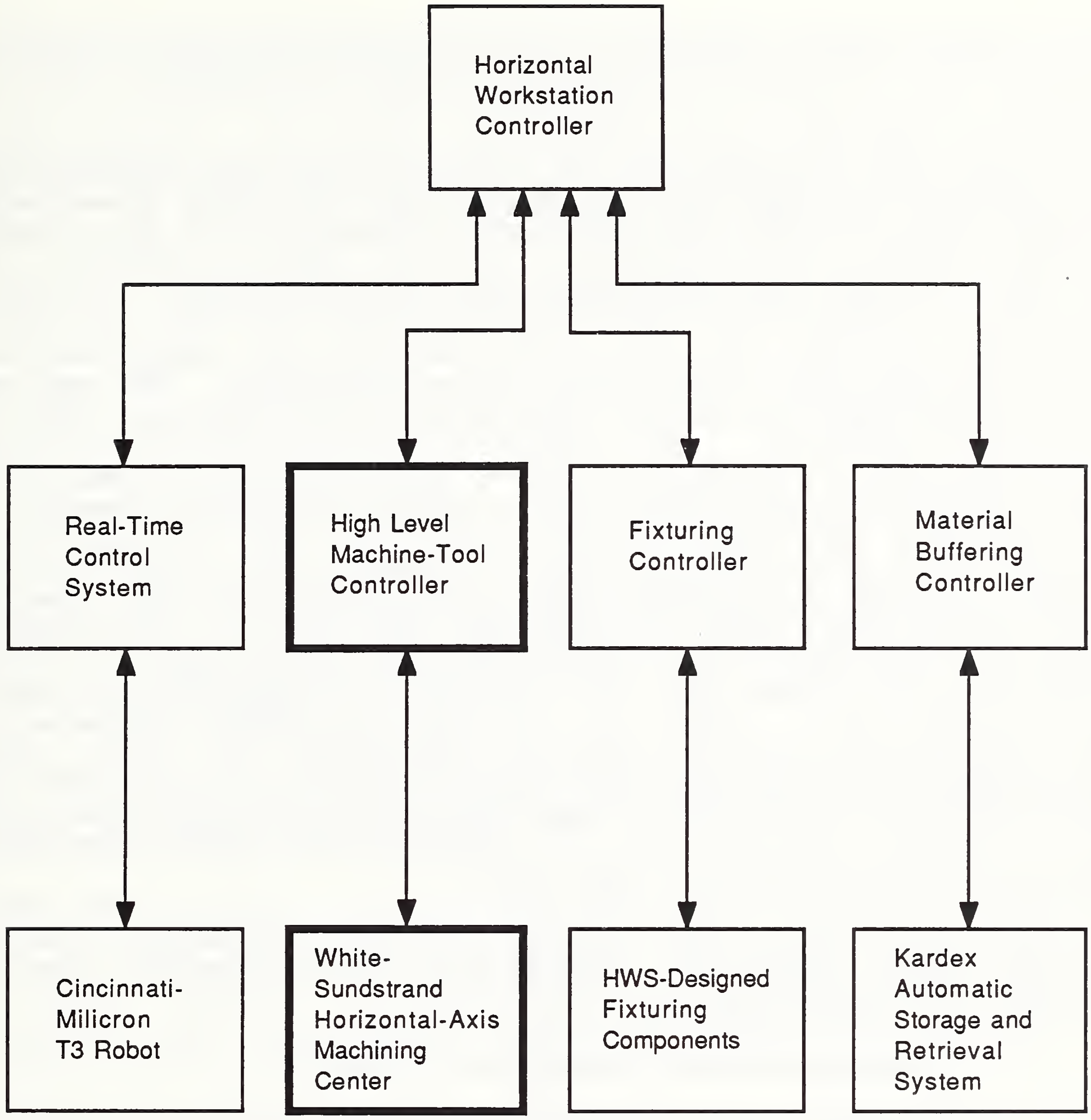


Figure 1 The High Level Machine-Tool Controller In The Horizontal Workstation

these equipment components has its own control system. Workstation tasks are performed through coordination of these equipment components by the Horizontal Workstation Controller.

Configuring the workstation, at the time of this documentation, means loading the tool drum with the tools required to machine a specific part. The Horizontal Workstation Controller (HWSC) commands the Material Buffering Controller (MBC) to present trays for robot access when unloading and loading tools. It commands the HLMC to rotate the tool storage drum for robot access to tools or to vacant tool slots. It commands the robot controller (RCS) to load and unload tools from the drum. And the HWSC commands the HLMC to release old tools and to grasp new tools.

When commanded to machine a part[a], the HWSC first commands the MBC to present a tray with a part blank for robot access. It then commands the HLMC to move the pallet with the appropriate fixture for robot access, and it commands the HWS Fixturing System to open the fixture jaws for part insertion. The HWSC then commands the robot to transfer the part to the fixture, coordinates the HWS Fixturing Controller and the RCS in the task of inserting the part into the fixture, and commands the HLMC to machine the part. When the machining has been completed, the HWSC commands the HLMC to position the pallet for robot access, and manages the robot and the fixturing system during removal of the part from the fixture, commands the MBC to present a tray with a vacant location for the machined part, and commands the robot to store the part in the tray.

The HLMC operations during these tasks are described throughout this document. Refer to sections II.3.1 and IV.4.2 for more detailed task descriptions.

3. REQUIREMENTS FOR AUTOMATED OPERATION

The HLMC, like all equipment components of the Horizontal Workstation, is a real-time sensory interactive control system. It has the following external interfaces: command and status interfaces to the control level above (HWSC), command and status interfaces to the equipment level below (White-Sundstrand computer), and command and status interfaces to a factory-wide data base (IMDAS). The HLMC updates information in the

---

[a] This is a simplified description of the part machining task, and is only meant to convey the sense of general workstation operations. Refer to HWSC documentation [10], [11], [12] for a more detailed description of workstation tasks.

interfaces to the upper and lower control levels in real time, and it can retrieve control data in real time.

Additionally, the HLMC retrieves and responds to sensory information and updates local control data in real time. Also, the tasks that the HLMC executes are generic, in that they require the part-specific data to be accessed in real time.

3.1 Interface To The Workstation Controller

The interface between the HWSC and the HLMC was designed to provide the workstation controller with a high level of control of the machine tool controller when coordinating equipment components to perform workstation tasks. The HWSC gives the HLMC high-level commands that the HLMC decomposes into a series of low-level commands to send to the White-Sundstrand computer.

To configure the workstation before making a part, the HWSC coordinates the HLMC with other equipment components of the workstation to load the White-Sundstrand's tool drum with the tools needed to machine that part. The configuration commands that the HWSC sends the HLMC cause it to rotate the tool drum and to grasp or release a tool.

To make a part, the HWSC commands the HLMC to position a pallet for robot access to a fixture and to execute the part programs required to machine the part.

When positioning a pallet, the HLMC sends the White-Sundstrand's computer the necessary combination of lower level machining center commands to move the pallet. The HLMC detects completion of these commands by reading status from the White-Sundstrand's computer and the outputs from the sensors on the machining center, summarizes this status, and sends it to the HWSC.

To machine a part, the HLMC uses parameters appended to the HWSC-to-HLMC command string to determine the part programs to run and the appropriate programs to execute. The HLMC then sends a series of lower level commands to the machining center so that it executes the part programs to machine the part. Again, the HLMC verifies the success of these commands by reading the status from the White-Sundstrand and the outputs from the sensors on the machining center, and summarizes the status for the HWSC.

The HWSC, therefore, need only be concerned with machining center tasks at the highest level, while the HLMC decomposes those tasks and performs the lower level steps required to perform them. The HLMC summarizes the lower level status from the machining center

into a more concise, general status that is appropriate for the HWSC.

3.2 Data-Driven Task Execution

The HLMC executes generic commands from the HWSC, with the task-specific information factored out as parameters to the command strings. This allows the HLMC to separate data from control. For example, all commands to machine a part are of the form:

EXECUTE [part geometry name]

There are many different ways the HLMC may machine a part. Some parts may require execution of only one part program, while others may require execution of several part programs. Still others may require execution of one or more part programs followed by a local inspection.

The HLMC uses the part geometry name as a pointer when retrieving more detailed information from a data base about how to machine a part. This information can be prepared and stored locally at the HLMC level, prepared locally and uploaded to the factory-wide data base, or prepared by an off-line CAD system and stored in the factory-wide data base. Regardless of the fashion by which the data is stored in a data base, it is retrieved in real time.

3.3 Update Of World Model

The HLMC was designed to use and update world-model information in real time. An example of how the HLMC can use the most current world-model information is its capability to use remote data to machine a part. If it is determined by an inspector or another inspection workstation in the factory that a horizontal workstation part is not being machined with the correct tolerances, machining center data for the part may be altered by an off-line programmer or programming system. The next time the HLMC is commanded to machine that part, it can retrieve the new, updated data in real time from the remote data base.

An example of how the HLMC updates world-model information about the White-Sundstrand is in its maintenance of a file of program names stored in the White-Sundstrand's buffer. Each time the HLMC executes a command to make a part it records in its local data base the names of the programs that are currently stored in the White-Sundstrand's buffer. The HLMC may use this file later to determine the programs to download to the White-Sundstrand.

Thus, the HLMC retains current information regarding the programs stored in the White-Sundstrand's local memory.

Long-term goals for the HLMC include updating other information in a data base regarding the physical configuration of the machining center, such as the tools stored in the tool drum and the amount of time that each of those tools has been used.

3.4 Programmability

There are interactive tools in the HLMC for adding new part data to its local data base. Control data lists may be added or changed. Part programs that were developed or altered at the machining center level may be uploaded to the HLMC. This data can then be uploaded to the factory-wide data base.

As mentioned above, new data can also be entered directly into to the factory-wide data base by another programmer, such as one using a CAD system, without any reprogramming at the HLMC level.

4. HARDWARE

The machine-tool equipment component of the Horizontal Workstation consists of a commercially available machining center and the HLMC computer system. The machining center is a White-Sundstrand horizontal-axis milling machine that has been modified to communicate with the HLMC. The HLMC is a microcomputer system with serial, parallel, and network communications boards for communicating with the machining center, the HWSC, and the factory-wide data base.

4.1 White-Sundstrand Horizontal-axis Omnimil

The White-Sundstrand machining center is a three and one-half axis horizontal-axis milling machine. Its tool storage system is a thirty slot tool drum. Tools are held to the tool drum with "tool fingers". Its two pallets slide on mechanical ways on an axis in front of the machining center so that each may be positioned in front of the spindle for part machining or to one side of the machining center for part fixturing.

The White-Sundstrand is equipped with a MicroSWINC computer numerical control (CNC) system to control the machine and its peripherals, such as the tape drive and tape punch. The MicroSWINC provides software capabilities for full contour machining functions and allows both automatic control of a

machining process and manual generation of machining instructions.

The MicroSWINC's executive software, the Machine Applications Program (MAP), and part programs are stored in its non-volatile bubble memory. The MicroSWINC also has volatile buffer memory where part programs and tool compensation offsets are stored and edited, and where part programs reside when they are executed.

The MicroSWINC may be connected to a distributed numerical control (DNC) system so that programs can be downloaded into and executed from its buffer storage. This link is used by the HLMC when it downloads programs to the White-Sundstrand.

Programs may also be entered into buffer storage using the MicroSWINC's paper tape reader. The MicroSWINC may be connected to its peripheral tape punch unit to record programs developed or edited locally. Programs are uploaded from the White-Sundstrand to the HLMC through this punch port.

4.2 Machining Center Modifications

Modifications to the machining center were made so that in an automated mode, it would perform the functions of a manually operated commercial White-Sundstrand product. That is, the HLMC would send the MicroSWINC computer the same keystrokes that an operator would type manually at the White-Sundstrand control panel to perform any machining center operation.

To determine the MicroSWINC communications protocol, a serial analyzer was connected to the link between the White-Sundstrand CRT and the MicroSWINC. While a manual operator executed machining center functions at the machine's keyboard, both sides of the communications were monitored and recorded.

An attempt was then made to emulate the keystrokes the operator manually typed. The keyboard half of these software conversations between keyboard and host computer was carried out using the serial analyzer and later the HLMC development system. This verified that a software emulation was indeed possible. A permanent serial link was then built connecting the White-Sundstrand to the HLMC.

In manual mode, an operator commands the machining center using the CRT on the control panel and by pressing buttons on the control panel, such as CYCLE START and FEED HOLD. In addition, the MicroSWINC activates indicator lights on the panel that convey status information of a machining center function to an

operator. Parallel communications lines from the HLMC are tapped into these same indicator lights and function buttons on the control panel. The HLMC can use that link to observe those indicators of the operating condition of the machining center and to initiate parallel commands of its own.

It was also necessary to add various devices and sensors to the machining center to support automated functions such as tool changing operations. The HLMC uses the parallel communications link to read those sensors and actuate those devices.

In addition, the manual tool changing position was modified to for robotic tool changing. A locating plate was added for robot positioning, and a computer operated pneumatic device was added so that the tool fingers at the tool changing position could be opened and closed remotely.

4.3 HLMC Computer Hardware

The HLMC computer resides in a ten-slot Multibus backplane bucket. Its Intel 8086-based computer board has 64 Kbyte on-board dual-ported RAM. An additional 512 Kbyte RAM expansion board serves as a common memory interface between the HLMC and the AMRF network. There is also an AMRF network board that interfaces with the HLMC common memory.

For serial communications, there are two Intel iSBC-534 boards supplying a total of eight serial ports. The HLMC Forth-based multitasking operating system uses these ports for terminal and background tasks.

For parallel communications, the HLMC uses an Intel SBC-519 parallel expansion board. This board provides up to 72 bi-directional I/O lines for sending or receiving TTL signals. The SBC-519 board interfaces with two Intel iCS-920 Digital Signal Conditioning/Termination Panels that each contain twenty-four two-wire input and output channels with sockets for optically isolated interface modules that operate within a variety of voltage ranges.

The HLMC's mass storage device is a 20 Mbyte Winchester disk. The disk is controlled by a RIMFIRE 38A disk controller board.

The HLMC also has a TCU-410 board that provides a battery-backed non-volatile clock/calender.

III. ARCHITECTURE

1. WORKSTATION COMMAND AND STATUS

The command and status interfaces between the HLMC and the HWSC meet the protocol requirements that exist for all WSC-to-equipment controller interfaces in the HWS.

The HLMC begins execution of a new command when the command number field of its command interface is incremented. The command names from the HWSC are generic. All the part-specific data is included in parameter fields following the command.

The HLMC sends a status message to the HWSC each time its status changes. A status message contains an echoed command number field that the HWSC compares to its current outgoing command number. With this information - command number and echoed command number - the HWSC can determine whether or not HWSC commands are synchronized with HLMC status. Status messages, like commands, are generic, with task-specific information factored out as parameters.

2. DATA BASE

The HLMC may access a local and a remote data base. It can operate using both data bases - accessing from the remote data base when local data for a part is not available - or it can access data from one or the other exclusively. Typically, when using a remote data base, the HLMC updates its local data. That is, any information the HLMC receives from a remote data base is subsequently stored locally so that the next time it is needed, a remote data base access will not be necessary.

In addition, the HLMC performs real-time updates of a local data file that has a record of the part programs that are stored in the White-Sundstrand's buffer.

2.1 Local Data Base

There are three types of data that the HLMC needs to access and update: part programs, resource lists, and the list of part program names that are stored in the White-Sundstrand's buffer memory. The HLMC uses four files to store this data.

2.1.1 Part Programs - Local Storage

There are two files that the HLMC uses to store part programs locally. One is an index of the programs that are stored and the other contains the programs themselves.

The index of part program names also contains the time and date the program was stored, as well as its location on the local disk and its length.

The part programs are stored in a file that is managed with the Buddy Management System [7, 9]. This file management system uses an algorithm to allocate and to "deallocate" contiguous blocks of disk space and eliminates the need for garbage collection.

If the HLMC attempts to store locally a part program that has the same name as another that is already on file, it overwrites the old version with the new version. When programs are either uploaded from the White-Sundstrand or downloaded from a remote data base, it is assumed that the local programmer or the higher level programming system (that manages data for the entire factory) is intentionally overwriting a previous version that may exist.

2.1.2 Resource Lists - Local Storage

Resource lists - the lists of the names of part programs required to machine parts - are stored locally in one file. Each resource list may contain up to ten part program names. These programs may be for certain features of a part or for inspection sequences.

In an automated mode, resource lists are written to the file once. A resource list for a part will never change. If the part is to be modified by adding or deleting a certain feature, its resource list is changed and is given a new name. (If it becomes necessary to replace a resource list with another of the same name but with different contents, as in a case when a resource list's contents have been corrupted, the list may be deleted manually and restored as it was intended.

2.1.3 Part Programs in the Machining Center's Buffer

The HLMC keeps a file of the names of the programs that are stored in the White-Sundstrand's buffer. The file, which is strictly local, is updated in real time before the HLMC begins

machining a part. The purpose of recording this information is described below.

The HLMC does not need to download a part program to the White-Sundstrand if the program is already available in the White-Sundstrand's buffer. To avoid any unnecessary downloads, the HLMC copies the White-Sundstrand's buffer directory into a file before machining each part, and compares the part's resource list to the list of programs that are already in the buffer. The HLMC then begins execution of the sequence of part programs on the resource list. If any of the programs on the resource list are not already in the White-Sundstrand's buffer, the HLMC downloads it immediately before execution.

Keeping this information in a file saves the HLMC execution time in cases where a part's resource list contains more than one part program. To read the buffer directory, the HLMC must initiate a series of communications handshakes with the White-Sundstrand. By reading this information once and storing the results in a file, the HLMC does not have to repeat those communications handshakes to read the buffer directory before executing each program on the resource list.

2.2 Remote Data

During automated operation the HLMC may retrieve its control data from a remote data base. All data retrieved from the remote data base is stored locally after the retrieval. This avoids repeated retrievals of the same data.

When the HLMC is using remote data and it receives an HWSC command to machine a part, the HLMC retrieves all the necessary data. If the resource list is not already stored locally, the HLMC retrieves it first from IMDAS. The HLMC then retrieves each part program from the data base and stores it locally. If a part program from the remote data base is already on file in the HLMC's local data base, the local program is overwritten with the remote program. The assumption here is that all data in the remote data base is the most current.

Programs and resource lists may be uploaded to IMDAS from the HLMC's local data base when the HLMC is not running in an automated mode. This is a manual operation, and its status is monitored by reading the remote data base's status mailbox.

3. MACHINING CENTER INTERFACES

A person operating the White-Sundstrand manually commands it by typing at the machining center's keyboard terminal and pressing buttons on the control panel. Similarly, the operator determines the White-Sundstrand's status by reading messages on the CRT and looking at indicator lights on the front panel.

There are two interfaces between the HLMC and the White-Sundstrand, a serial and a parallel interface. These interfaces allow the HLMC to emulate the functions performed by an operator who uses the control panel. The HLMC uses the serial interface to send the White-Sundstrand's computer the same information that an operator sends by typing at the machining center's keyboard terminal. The parallel interface is the interface to the buttons and indicator lights on the front panel, and in addition, to the sensors and mechanical actuators that were added to the machining center for automated operation. The information passed through both of these interfaces make up the command and status messages that are passed between the HLMC and the White-Sundstrand.

3.1 Serial Interface

An RS-232 serial link to the HLMC is connected in parallel with the communications link between the terminal on the machining center's control panel and the MicroSWINC computer (Figure 2). In automated mode (as pictured), the HLMC emulates the machining center terminal. To command the White-Sundstrand, the HLMC sends the MicroSWINC the same characters that are transmitted when an operator types at the terminal on the control panel. To interpret status, the HLMC retains a duplicate of the White-Sundstrand's terminal screen in RAM and monitors the status messages that the MicroSWINC sends the terminal screen.

In addition to the explicit commands and status messages that pass between the operator terminal and the MicroSWINC, there are also communications handshaking messages that are passed between the two that do not appear on the screen. For example, the MicroSWINC sends the terminal screen some unprintable control characters for which it expects to receive a response. When emulating the terminal during automated operation, the HLMC also emulates these responses.

3.1.1 Serial Commands

When the HLMC receives a command from the HWSC to perform a high-level function, it decomposes the command into the low-level commands that are to be sent to the machining center's computer.

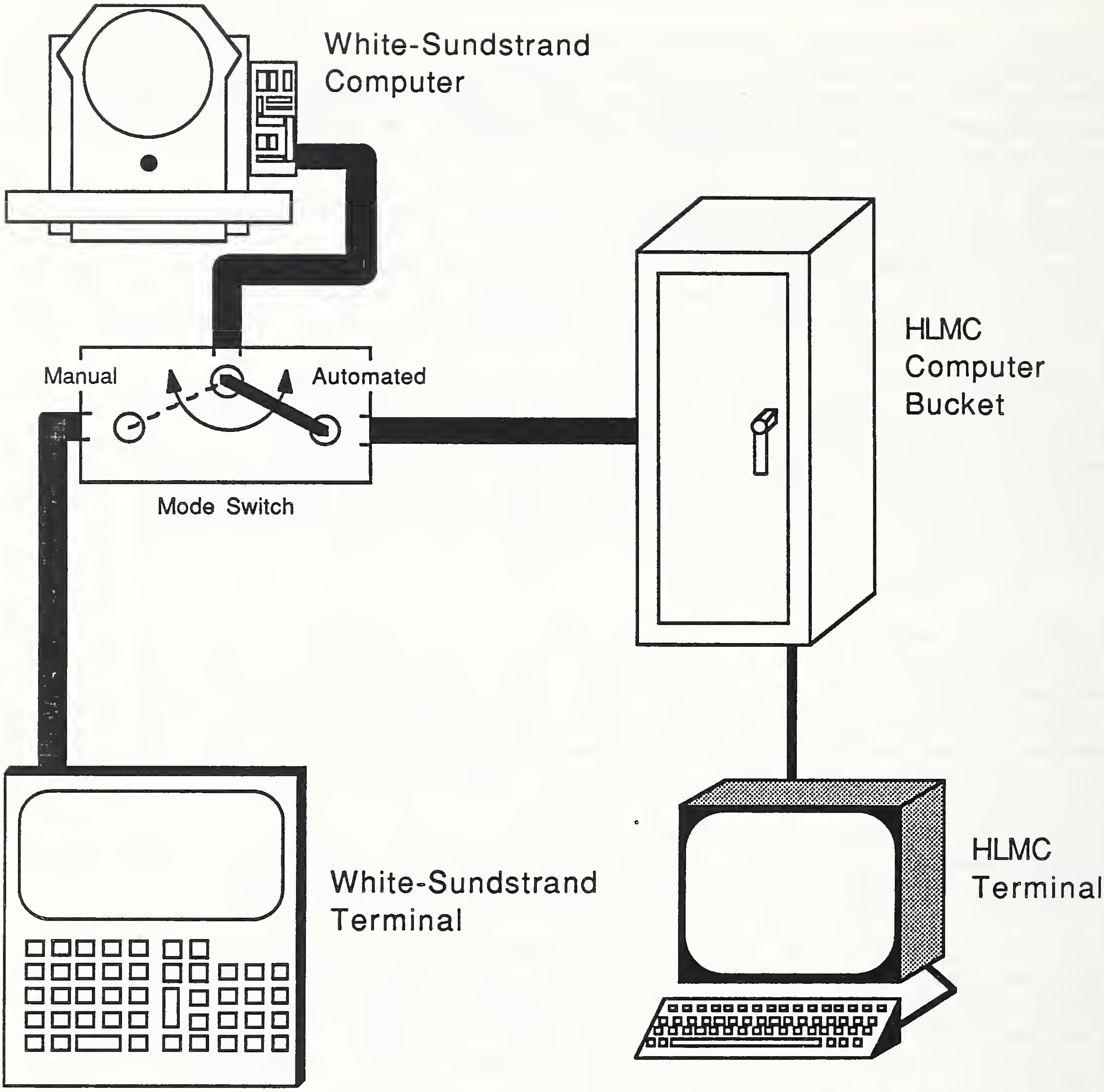


Figure 2 HLMC Serial Interface To The White-Sundstrand
In automated mode, the HLMC emulates the
White-Sundstrand terminal.

These commands are assembled as output strings in a buffer, the necessary parameters are filled in, and the string is transmitted to the MicroSWINC.

The serial commands transmitted to the MicroSWINC computer are streams of ASCII characters. Each command has a maximum length of thirty-two characters - the same length as the command strings an operator types at the machining center's terminal.

3.1.2 Serial Status

An HLMC task receives all characters that the MicroSWINC sends to its terminal. Most of the characters that the MicroSWINC sends are printable ASCII characters. However, some of the characters are control characters that direct the terminal to alter its cursor position. (Some of these cursor directives are HOME, LINEFEED, and CLEAR.) The HLMC has been programmed to recognize these control characters and to translate them into commands that alter its internal screen buffer's cursor position. The incoming characters that are not cursor positioning commands are ASCII display characters and are stored in the buffer.

The HLMC determines the White-Sundstrand's status from the messages that the MicroSWINC sends its terminal screen. To verify that the MicroSWINC is executing or has completed execution of a command successfully, the HLMC examines its internal image of the machining center's terminal screen. After it issues a serial command to the MicroSWINC, the HLMC monitors a portion of the terminal screen, waiting for a status response. The HLMC does not need to monitor the entire screen, for status is usually updated on a small portion of the display. For example, after the White-Sundstrand completes execution of a "manual data" (MD) type command, the only change in the terminal's display is that the message "END" appears on the 14th line. (Figure 3). Therefore, after issuing a manual data command, the HLMC need only monitor the 14th line of its internal representation of the screen instead of repeatedly interpreting the entire screen.

The HLMC has been programmed to recognize a successful response to every command that it may issue to the MicroSWINC. It has also been programmed to recognize error messages. For safety reasons, if a preprogrammed status message does not appear on the terminal, the HLMC stops White-Sundstrand execution and reports status to the HWSC.

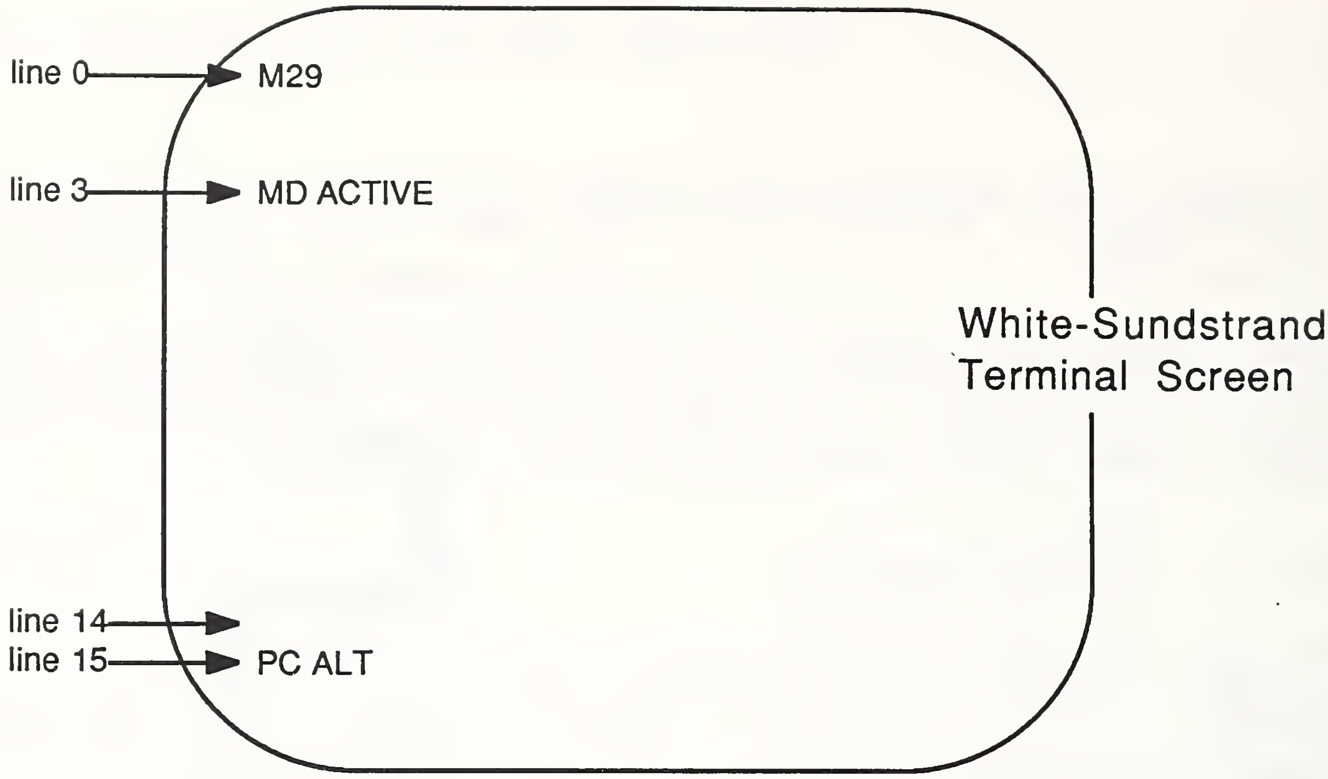


Figure 3A While the White-Sundstrand is executing a Manual Data command, "MD ACTIVE" is displayed on line 3 of its terminal screen.

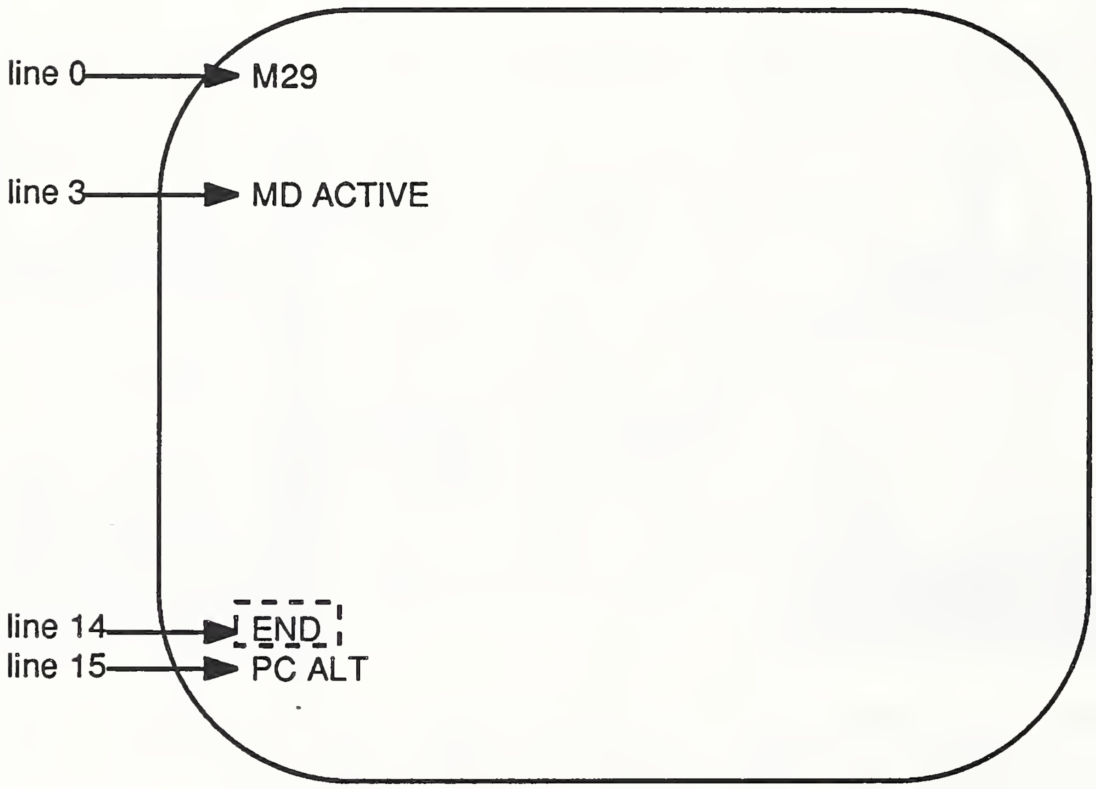


Figure 3B When the White-Sundstrand completes execution of a Manual Data command, it displays "END" on line 14 of its terminal screen.

Figure 3 White-Sundstrand Status Screens

After the HLMC issues a command to the White-Sundstrand over the serial interface, it determines status by monitoring the White-Sundstrand's terminal screen.

3.2 Parallel Interface

The configuration of the parallel communications system of the machining center controller is illustrated in Figure 4. The HLMC has a software interface (the HLMC Parallel Communications Module) to a parallel I/O board that resides in the HLMC bucket. This board is linked to a panel of solid-state output and input relays that are used to send and receive voltage signals to and from the machining center.

Each time the HLMC sends the machining center any command using the parallel communications link, it verifies that the command has been received and is being executed. This verification is typically transmitted over the status part of the parallel communications link. For example, to initiate execution of a part program, an operator presses the CYCLE START button on the control panel. While the part program is executing, an indicator light on the panel remains illuminated. When the program is completed, the indicator light is turned off. In automated mode, the HLMC activates the CYCLE START button. It then monitors the status of the process by monitoring both the terminal screen and the input module that indicates the status of the CYCLE START indicator light. If the CYCLE START light is turned off and if there are no error messages on the terminal screen, the HLMC interprets the status from the machining center for that program's execution as "DONE".

To use the parallel communications link to check the status of a process, the HLMC requests the required sensory information. The receipt of sensory information will never interrupt another HLMC process.

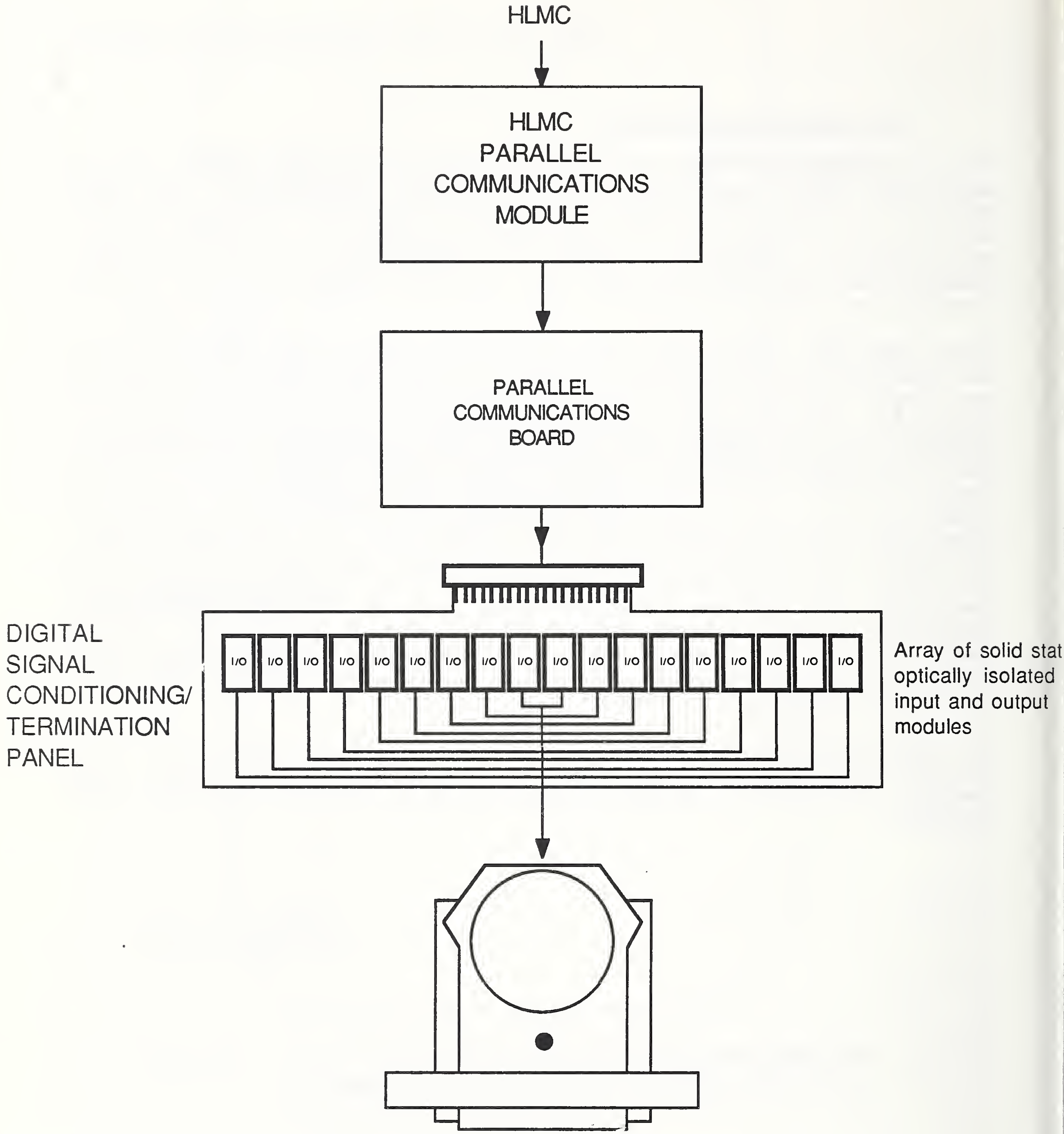


Figure 4 HLMC Parallel Communications Interface To The White-Sundstrand

IV. HLMC IMPLEMENTATION

1. MACHINING CENTER INTERFACES

For both the serial and parallel interfaces, there are several levels of software modules which allow a programmer to redesign parts of the interfaces with minimal effects on the other modules. The specific hardware chosen for these tasks may also be replaced, requiring software modifications only at the lowest level.

That is not to say that swapping the White-Sundstrand with another machining center would require only minimal communications hardware changes at the lowest level. Unfortunately, manufacturers produce controllers with such widely varying characteristics that it is impossible to standardize these interfaces. In the case of the White-Sundstrand, the effects of the MicroSWINC control system language affect many levels of the controller, as well as the lowest level communications interface. However, if the White-Sundstrand in the HWS were to be replaced by another machining center with a MicroSWINC controller, but with different communications protocols, this interchange of machining centers might be done with minimal effect on the hardware and software.

Another advantage of the software modularization is that it has decreased the development time and led to successful results by allowing complete testing of each new capability from its lowest level functions to its highest level implementation. The use of Forth, a highly interactive programming language, has also been a major factor in the development of the HLMC. A discussion of the use of Forth in HWS control systems is in the Horizontal Workstation Controller documentation [12]. References will be made to executable Forth words and modules throughout the remainder of the document.

1.1 Serial Interface

The functions of the serial communications interface to the MicroSWINC controller are provided by a high-level Forth word called SDD.TASK. SDD.TASK is a continuous loop background task that sends and receives messages to and from the machining center. HLMC command modules build messages in an output buffer and set a flag to notify SDD.TASK of the outgoing message, and status modules examine the most recent screen image that was stored by SDD.TASK. These command and status modules do not have to be concerned with the mechanics of getting the messages to and from the MicroSWINC.

1.1.1 External Communications

The definition of SDD.TASK may be represented as:

```
: SDD.TASK  BEGIN
               SDD.CHAR.TO.INTERPRET?
                 IF  INTERPRET.SDD THEN
               SDD.MESSAGE.TO.SEND?
                 IF  SEND.MESSAGE.SDD THEN
            END ;
```

The cycle SDD.TASK consists of the series of words between BEGIN and END. SDD.TASK first checks for new inputs and then checks for new outputs. The word SDD.CHAR.TO.INTERPRET? checks if the MicroSWINC has sent any new characters to its terminal that the HLMC should store in its internal image of the machining center's terminal display. If a new character has arrived, INTERPRET.SDD stores it in memory. SDD.MESSAGE.TO.SEND? checks if there is a new HLMC message to be transmitted to the MicroSWINC. If there is an outgoing message, SEND.MESSAGE.SDD transmits it. Note that the structure of the word SDD.TASK prevents the HLMC from transmitting a message at the same time that the MicroSWINC sends a message. This would initiate an unrecoverable error condition.

An interrupt service routine receives all the characters and cursor directives sent by the MicroSWINC, stores them in a buffer, and increments a pointer into the buffer. SDD.CHAR.TO.INTERPRET? checks that pointer. If it has been incremented, a new character has arrived.

INTERPRET.SDD looks at the characters in the input buffer. If the character is a control character, the cursor is moved or the screen display is manipulated. Otherwise, INTERPRET.SDD stores the ASCII value in the buffer.

Other HLMC modules read the internal representation of the screen that the SDD.TASK builds. The input buffer is read- and write-protected to prevent the HLMC from reading the screen when a new message is coming in. The HLMC will not attempt to send any message to the MicroSWINC unless it is certain that the MicroSWINC has received its last command.

1.1.2 Commands To The Machining Center

To command the machining center, HLMC modules build messages in the output buffer and set a flag to let SDD.TASK know that there is a message in its output buffer to be transmitted. The commands that the HLMC sends the MicroSWINC are those that an operator would type at the terminal of the control panel. Some of the MicroSWINC commands that the HLMC uses during automated operation and their functions are listed below in Table 1.

Table 1 Machining Center Commands That The HLMC Uses During Automated Operation

MicroSWINC Commands	Function
RS	Resets the SWINC
MD	Enters Manual Data Screen
EN	Ends Screen
PC A, PC B, PC ALT, M29	Positions a Pallet
M06Tx (where x = tool number)	Changes a Tool
LR, D/, S/, DN	Downloads Part Programs
SE	Executes Programs, Downloads
DI	Displays Stored Programs
PI	Displays Axis Position Indicators

1.2 Parallel Interface

A parallel communications software package was written for several HWS computer systems that use the same hardware as the HLMC. This general purpose software has been expanded to meet the specific requirements of the HLMC.

There are, at the lowest level, modules that use a natural language syntax to read or actuate specific sensors or devices. Higher level modules provide more complex functions or give a broader perspective of the sensory state of the machining center For example, at the lowest level, there are modules that read one sensor at the tool changing position of the machining center, such as TOOL-THERE?. At a higher level, to determine if it is safe for a robot to remove a tool from the tool drum, a higher level module called TOOL-UNLOAD-OK? uses several lower level modules such as TOOL-THERE? and FINGERS-OPENED?.

Below is Table 2A, a list of the indicators provided by parallel inputs, followed by Table 2B, a list of the functions provided by the parallel outputs.

Table 2A Parallel Communications Inputs

Pallet at Position A
Pallet at Position B
Cycle Start Indicator
Feed Hold Indicator
M12 Indicator
Tool Fingers Opened
Tool Fingers Closed
Tool Present at Tool Change Station

Table 2B Parallel Communications Outputs

Cycle Start
Pallet Ready - Cycle Start
Feed Hold
Open Tool Fingers
Close Tool Fingers

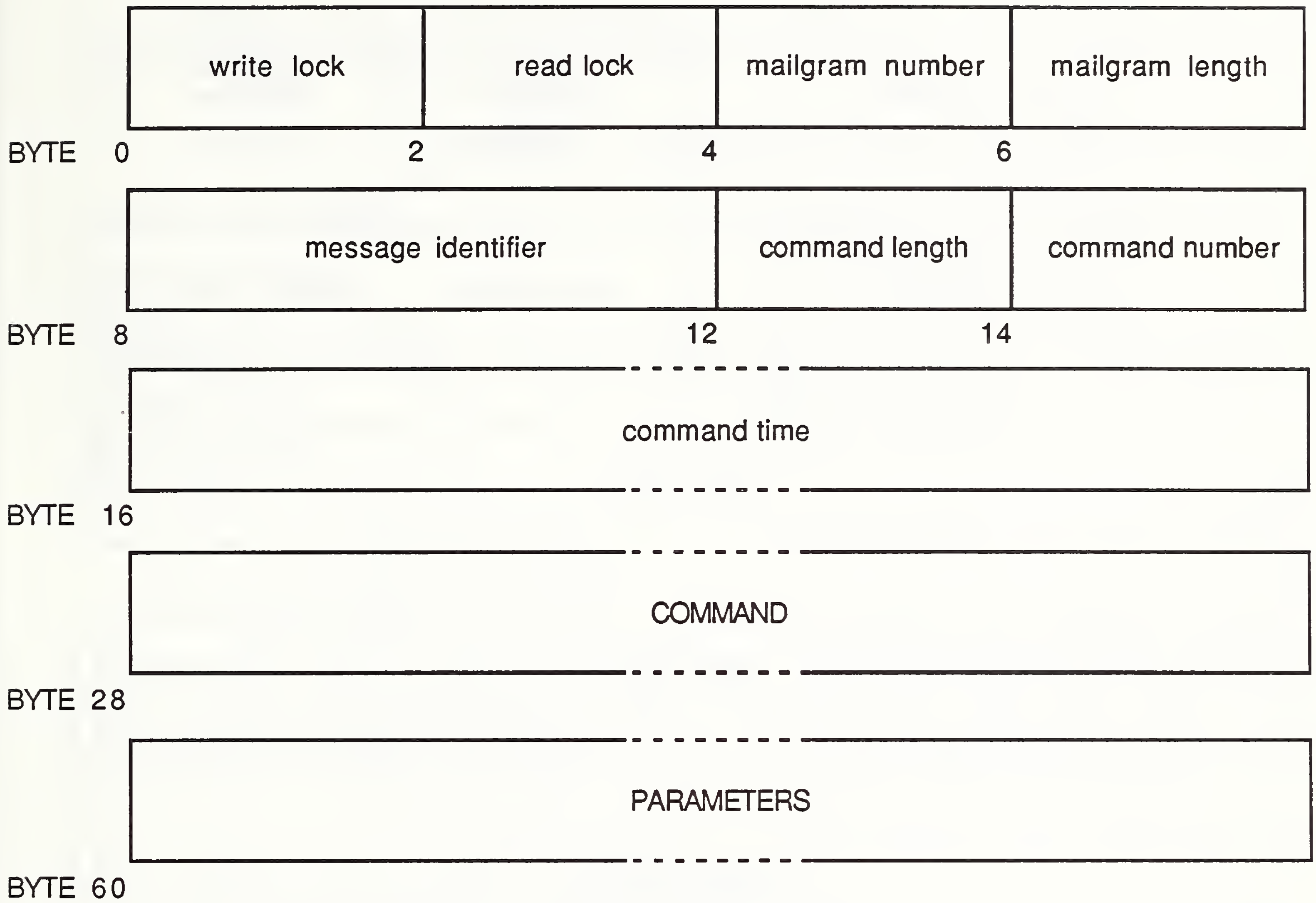
2. WORKSTATION CONTROLLER INTERFACE

The communications interface between the Horizontal Workstation Controller and the HLMC is an AMRF network interface. Figures 5A and 5B show the structure of the command and status mailboxes and include an explanation of each field.

The addresses in common memory of the first byte of each HWSC mailbox are given below in Table 3.

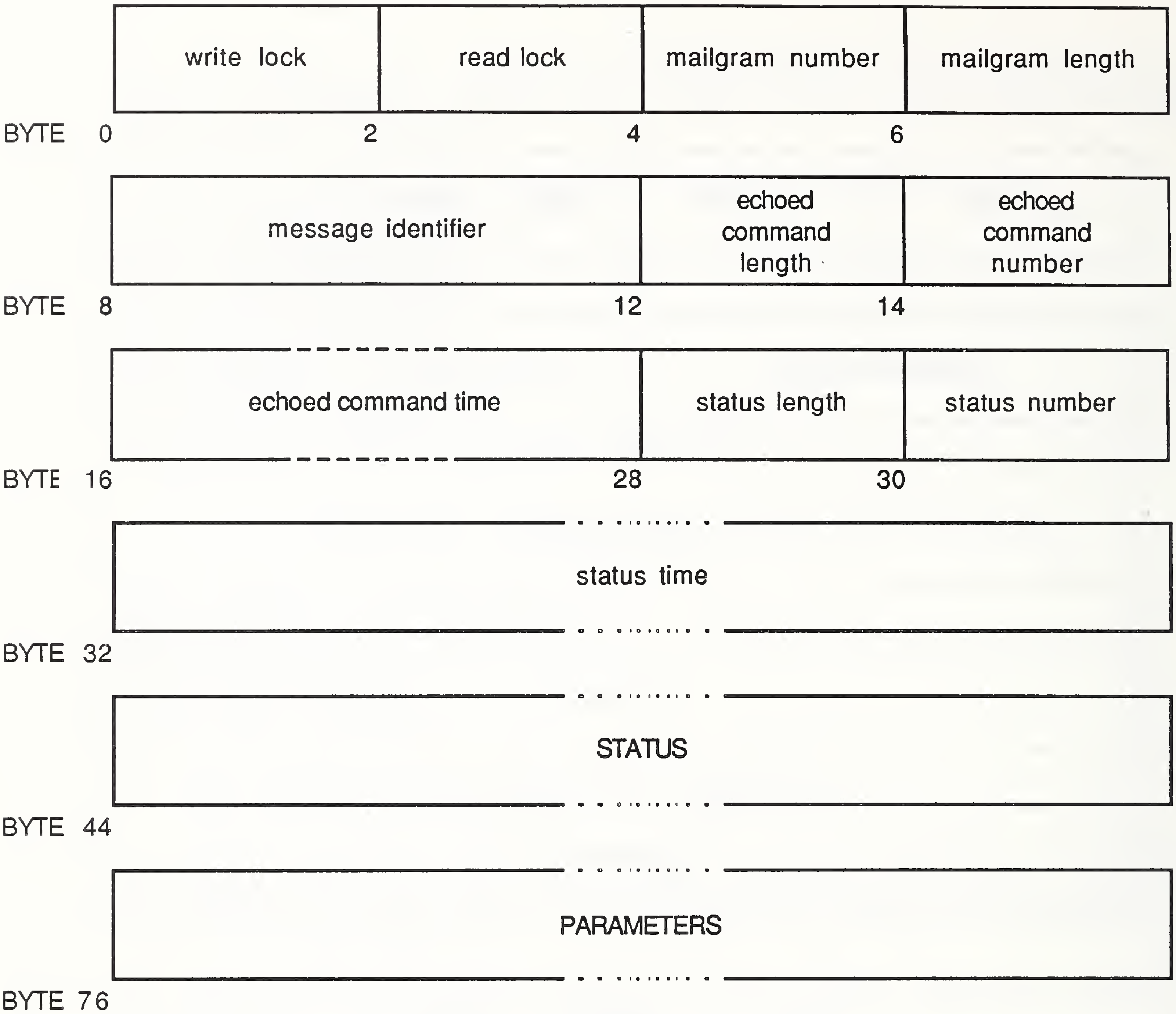
Table 3 Common Memory Addresses For HWSC Mailboxes

Mailbox	HLMC Memory Address (HEX)
Horizontal Workstation Command	2400
Horizontal Workstation Status	2300



Byte	Length	Field
0	2	Write Lock
2	2	Read Lock
4	2	Mailgram Number
6	2	Mailgram Length
8	4	Message Identifier
12	2	Command Length
14	2	Command Number
16	12	Command Time
28	32	Command
60	32	Parameters

Figure 5A Workstation Command Mailbox



Byte	Length	Field
0	2	Write Lock
2	2	Read Lock
4	2	Mailgram Number
6	2	Mailgram Length
8	4	Message Identifier
12	2	Echoed Command Length
14	2	Echoed Command Number
16	12	Echoed Command Time
28	2	Status Length
30	2	Status Number
32	12	Status Time
44	32	Status
76	16	Parameters

Figure 5B Workstation Status Mailbox

The commands that the Horizontal Workstation Controller can send the HLMC are shown below in Table 4. A high-level description of each of these commands is given in section IV.4.2.

Table 4 Commands Sent To The HLMC

Command (32 bytes)	Parameter (16 bytes)
POSITION TABLE	pallet location
EXECUTE	part geometry name
PRESENT TOOL-POCKET	serial number
GRAB TOOL	serial number
PRESENT TOOL	serial number
RELEASE TOOL	serial number
RESET TOOL-POCKET	
FEED HOLD	

3. DATA BASE INTERFACE

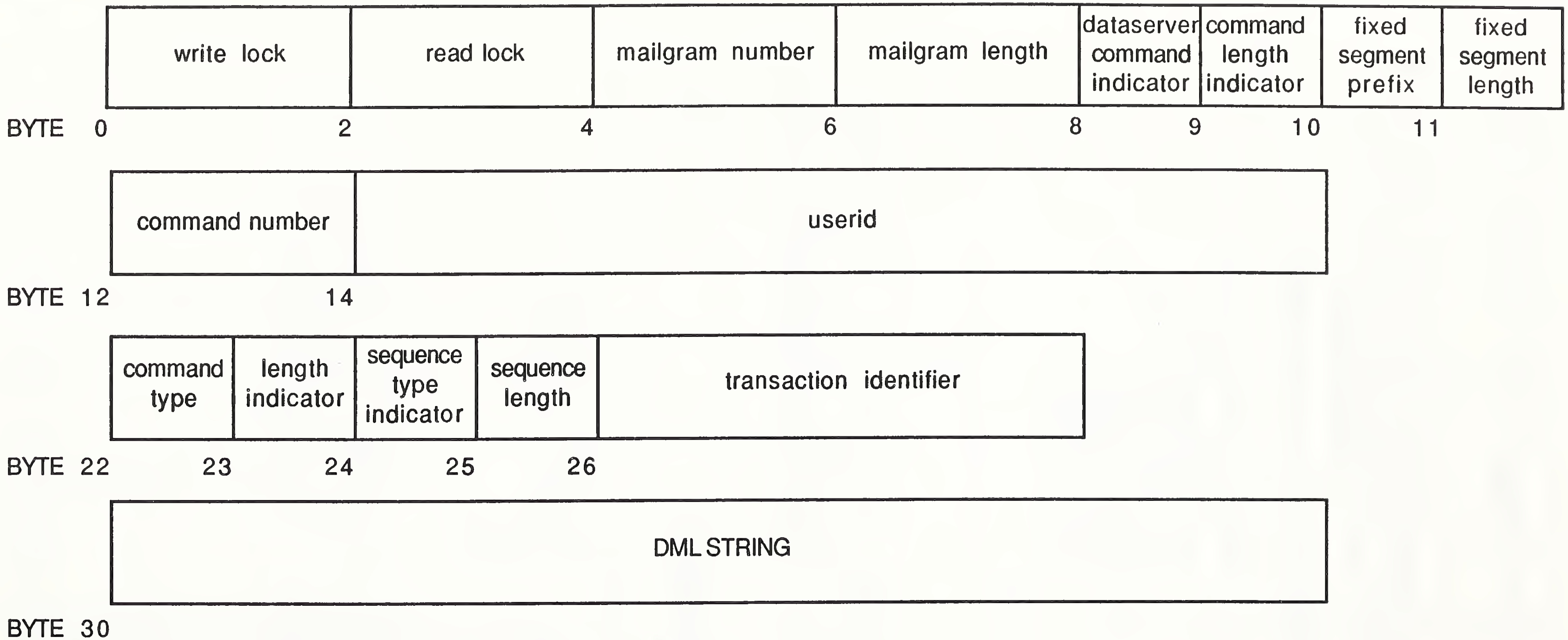
Figures 6A, 6B, and 6C show the structures of the command, status, and data mailboxes between the HLMC and the IMDAS, with each field identified. For a detailed explanation of the mechanics of this interface, refer to the IMDAS documentation [6, 8].

The addresses in common memory of the first byte of each data base mailbox are given below in Table 5.

Table 5 Common Memory Addresses For Data Base Mailboxes

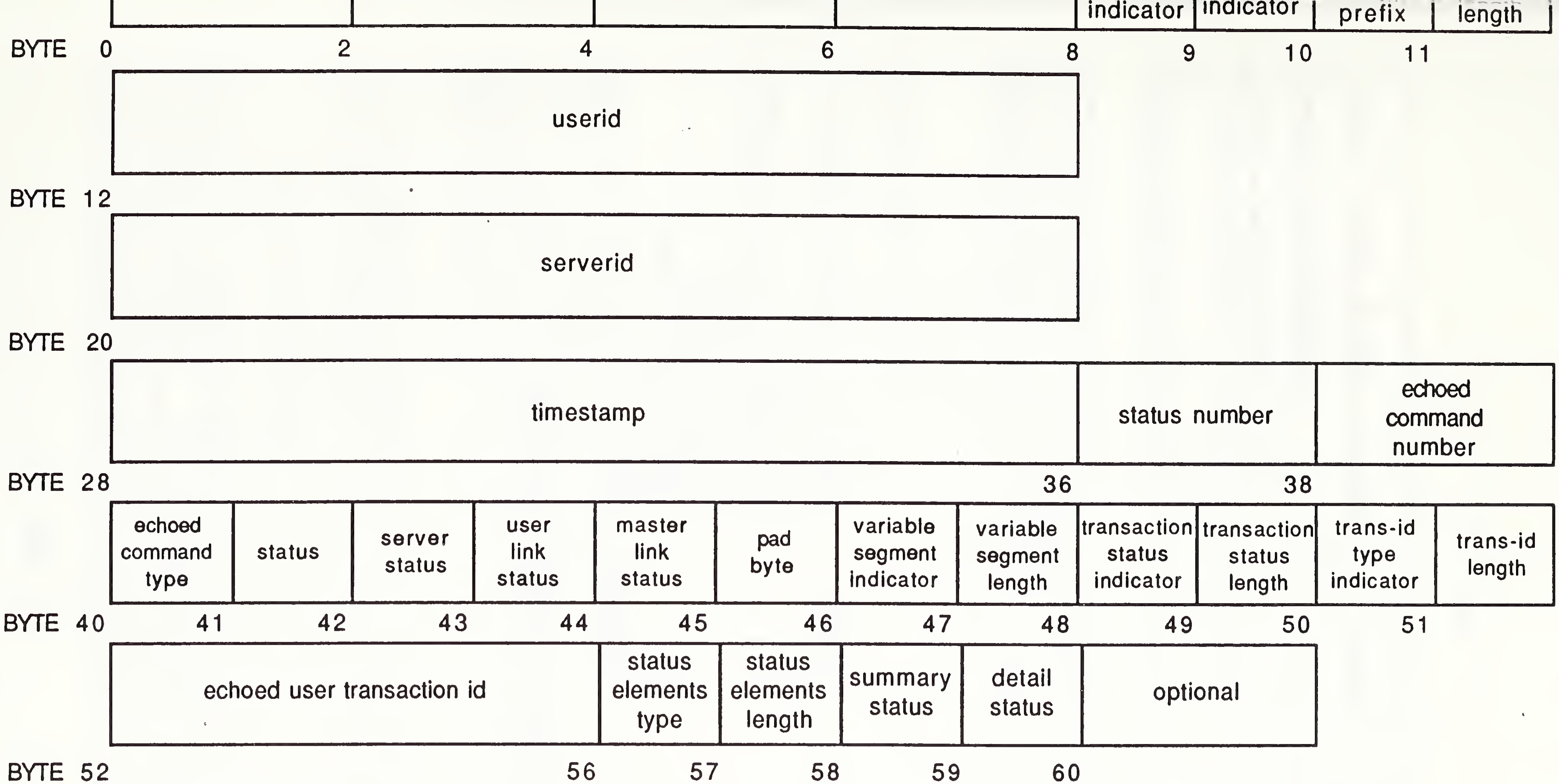
Mailbox	HLMC Memory Address (HEX)
Data Base Command	2500
Data Base Status	2600
Data Base Data	2700

During fully integrated automated operation, as opposed to standalone workstation operation, the HLMC only retrieves data from IMDAS. The two types of data that the HLMC retrieves are resource lists and part programs. The commands that the HLMC gives to the data base to retrieve this data are shown below in Table 6A.



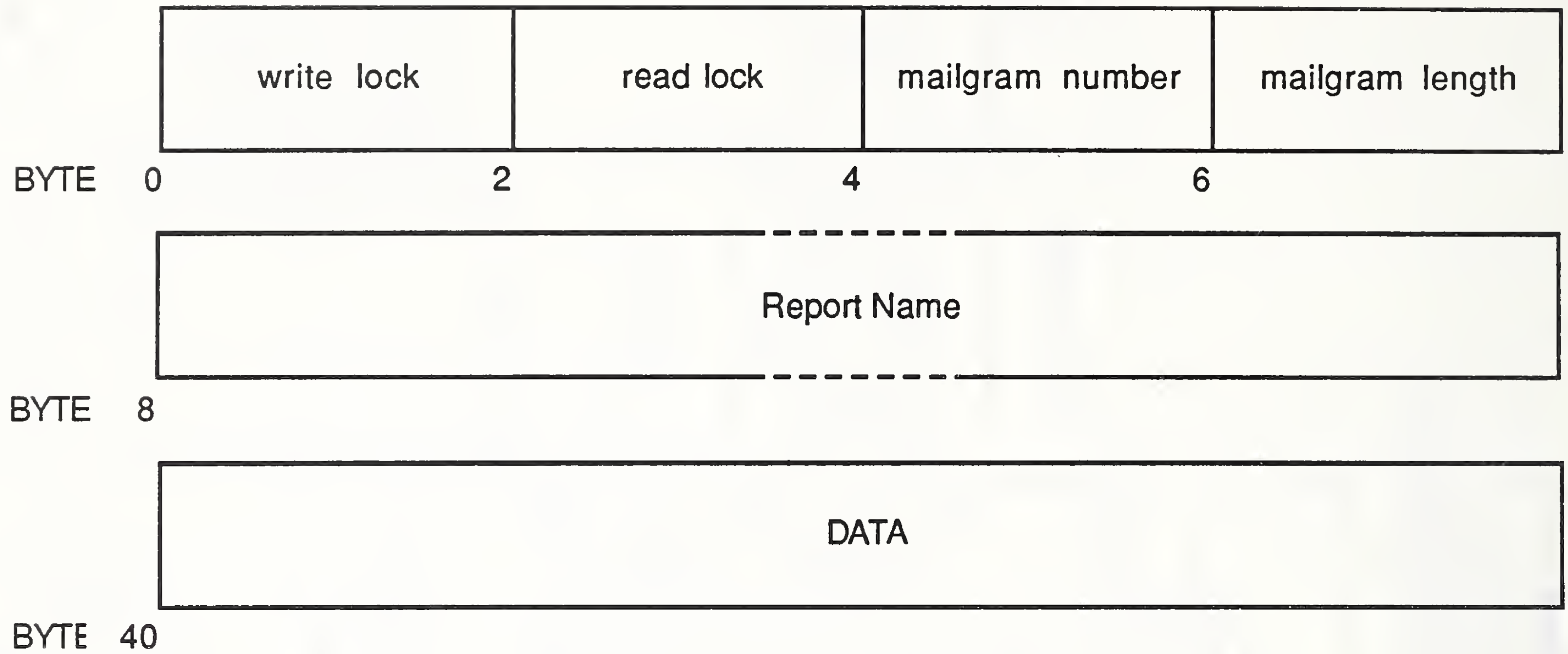
Byte	Length	Field
0	2	Write Lock
2	2	Read Lock
4	2	Mailgram Number
6	2	Mailgram Length
8	1	DataServerCommand Indicator
9	1	Command Length Indicator
10	1	Fixed Segment Prefix
11	1	Fixed Segment Length
12	2	Command Number
14	8	Userid
22	1	Command Type
23	1	Length Indicator
24	1	Command Sequence Type Indicator
25	1	Command Sequence Length
26	4	Transaction Identifier
30	variable	DML Command String

Figure 6A IMDAS Command Mailbox



Byte	Length	Field	Byte	Length	Field	Byte	Length	Field
0	2	Write Lock	36	2	Status Number	49	1	Transaction Status Length
2	2	Read Lock	38	2	Echoed Command Number	50	1	Transaction-id Type Indicator
4	2	Mailgram Number	40	1	Echoed Command Type	51	1	Transaction-id Length
6	2	Mailgram Length	41	1	Last Command Status	52	4	Echoed User Transaction-id
8	1	DataServer Status Indicator	42	1	Server Status	56	1	Status Elements Type
9	1	Status Length Indicator	43	1	User Link Status	57	1	Status Elements Length
10	1	Fixed Segment Prefix	44	1	Master Link Status	58	1	Summary Status
11	1	Fixed Segment Length	45	1	Pad Byte	59	1	Detail Status
12	8	Userid	46	1	Variable Segment Indicator	60	variable	Optional
20	8	Serverid	47	1	Variable Segment Length			
28	8	Time Stamp	48	1	Transaction Status Indicator			

Figure 6B IMDAS Status Mailbox



Byte	Length	Field
0	2	Write Lock
2	2	Read Lock
4	2	Mailgram Number
6	2	Mailgram Length
8	32	Report Name
40	variable	Data

Figure 6C IMDAS Data Mailbox

Table 6A Data Base Commands - Retrievals

Command and Parameter	Function
SELECT * FROM CONTROL_LIST WHERE ENDITEM = 'resource list name'	Retrieves Resource Lists
SELECT * FROM CONTROL_DATA WHERE ENDITEM = 'part program name'	Retrieves Part Programs

Off line, the HLMC may update the data base versions of its resource lists and part programs. The commands the HLMC uses to update this information are shown in Table 6B.

Table 6B Data Base Commands - Updates

Command and Parameter	Function
INSERT INTO CONTROL_LIST VALUES 'resource_list'	Uploads Resource Lists
INSERT INTO CONTROL_DATA VALUES 'part_program'	Uploads Part Programs

4. COMMAND EXECUTION

When the HLMC receives a new command (evident by a new command number) it executes DOER.COMMAND.CASE, a Forth case statement that compares the string in the command mailbox to the command strings shown in Table 4. DOER.COMMAND.CASE directs HLMC execution to the routine that performs the task. If a match with those strings is not found, the HLMC responds to the HWSC with a "CAN'T DO" message.

In this section, a description of one of the software modules to which DOER.COMMAND.CASE often directs execution is followed by a high-level description of each of the HLMC tasks.

4.1 Task Routines

The HLMC tasks have been broken down into task routines. Each routine, or software module, is the series of actions necessary to perform the task. Each action in the series is represented by a high-level Forth word that can execute independently. If an error occurs during any part of the task, the execution of the task is stopped and status is updated to the HWSC.

An example HLMC routine, one for the part machining task, is shown below.

```
: EXECUTE.DOER
     FETCH.RESOURCE.LIST.FROM.DB
     FETCH.NCPGMS.FOR.GEOM
     COMPLETE.PALLET.SHUTTLE
     DOWNLOAD&RUN.NCPGMS.FOR.GEOM ;
```

Each word in EXECUTE.DOER represents a major portion of the part machining task. If an error occurs during execution of any of the four words in EXECUTE.DOER, the task's execution stops and the HLMC sends an error message to the HWSC. Each of those four words consists of progressively lower level Forth words. If an error occurs at any lower level, execution is suspended and directed out of the higher level words.

4.2 HLMC Task Execution

Below are high-level descriptions of the steps that the HLMC performs after receiving commands from the HWSC. In these ideal cases, no errors occur. Note that the scenarios described can vary depending on many different variables. The cases below should, however, convey a general idea of the tasks' execution behaviors. When each task is completed, the HLMC reports status to the HWSC.

4.2.1 Positioning A Pallet

Command: POSITION TABLE pallet location

Purpose: The HLMC moves the specified pallet to its pallet loading position.

Before describing the HLMC's execution of the pallet positioning command, it is important to describe the machining center's pallet positioning modes.

During a pallet shuttle, the machining center moves a pallet from the "home" position to a pallet loading position, and then back to the home position. The machining center has three different pallet positioning modes. Two of the modes are for moving either one of the pallets from the home position to their loading positions and back to the home position. The third mode is for moving one pallet from the home position to its loading position while moving the other pallet from its loading position to the home position. For this discussion, the pallet that is shuttled

in front of the spindle will be referred to as the "active" pallet.

In the first pallet positioning mode, the pallet to the right of the spindle (Pallet A) is the active pallet. In that mode, Pallet A is shuttled between the home position and its pallet loading position. The second mode makes the pallet to the left of the spindle (Pallet B) the active pallet. The third mode alternates the active mode between the two pallets.

The purpose of the pallet positioning command is not only to position the appropriate pallet for robotic part fixturing, but also to make that pallet the "active" pallet for machining. When the HLMC receives the POSITION PALLET command, it reads sensors on the machining center to determine the positions of both pallets and it checks the pallet positioning mode. It then shuttles one or both of the pallets, if necessary, to the proper positions and it makes the specified pallet the active pallet.

4.2.2 Machining A Part

Command: EXECUTE part geometry name

Purpose: The HLMC executes the series of part programs required to machine the specified part geometry.

The HLMC first checks its local data base to see if the resource list for the specified part geometry is present in its local data base. If the resource list is not stored locally, the HLMC retrieves it from the remote data base.

Next, the HLMC retrieves any part programs listed in the resource list that are not stored locally from the remote data base. It then reads the directory of programs stored in the White-Sundstrand's buffer to determine which programs in the resource list, if any, need to be downloaded from the HLMC to the machining center.

The appropriate pallet is then shuttled in front of the spindle and each part program in the resource list is then downloaded (if necessary) and executed.

4.2.3 Rotating The Tool Drum - Loading

Command: PRESENT TOOL-POCKET serial number

Purpose: The HLMC presents a vacant tool pocket of the machining center's tool drum to the pallet changing position for robotic tool loading.

The HLMC rotates the tool drum so that the specified tool pocket is moved to the tool changing location. When the drum is positioned properly, the HLMC reads sensors to verify that the tool pocket at the tool changing position is unoccupied, and then it opens the tool fingers so that the robot can move a tool between the tool fingers. It verifies that the fingers opened fully before reporting status back to the HWSC.

4.2.4 Grasping A Tool

Command: GRAB TOOL serial number

Purpose: The HLMC grasps a tool that the robot has moved to the tool changing position.

The HLMC verifies that the robot is holding the tool so that the tool fingers will grasp the tool properly when they close. After sending the machining center the command to close the tool fingers, the HLMC reads sensors to verify that the fingers are closed on the tool and that the tool is being held in the proper position.

4.2.5 Rotating The Tool Drum - Unloading

Command: PRESENT TOOL serial number

Purpose: The HLMC rotates the tool drum so that a tool is in position for robotic tool unloading.

The HLMC rotates the tool drum so the specified tool pocket is presented to the tool unloading position. It reads sensors to verify that there is a tool in the tool pocket before reporting status to the HWSC.

4.2.6 Releasing A Tool

Command: RELEASE TOOL serial number

Purpose: The HLMC releases a tool that is at the tool changing position so that the robot can remove it from the drum.

The HLMC opens the tool fingers at the tool changing position. It then verifies with sensors that the tool fingers opened fully before reporting status to the HWSC.

4.2.7 Closing The Tool Fingers After An Unload

Command: RESET TOOL-POCKET

Purpose: The HLMC closes the tool fingers at the tool changing position following a robotic tool unload.

When the HLMC receives the RESET TOOL-POCKET command, it verifies that there is not a tool at the tool changing position, closes the fingers, and then verifies that the tool fingers have closed completely before reporting status to the HWSC.

4.2.8 Feed Hold

Command: FEED HOLD

When it receives this command from the HWSC, the HLMC immediately activates the machining center's "Feed Hold" button. This command was designed to be used during an emergency.

5. OFF-LINE DATA PREPARATION

When data is prepared locally (at the equipment level of the AMRF hierarchy rather than at a higher level), programmers perform the following steps:

1) An NC programmer writes the part programs for a part and loads them into the White-Sundstrand buffer memory.

2) An HLMC programmer retrieves the part programs from the White-Sundstrand and files them in the HLMC's local disk storage.

3) The HLMC programmer creates a resource list for the part at the HLMC's operator terminal and stores it in the resource list file.

To upload the data that was prepared locally so that future remote accesses will retrieve this data, the HLMC programmer performs two additional steps:

4) The programmer uploads the part programs for the part to IMDAS.

5) The programmer uploads the resource list for the part to IMDAS.

The HLMC has software tools that simplify the procedure for uploading programs from the White-Sundstrand to the local HLMC data base, uploading data to the IMDAS, and for storing and deleting part programs and resource lists locally. For a full description of how to use these tools for off-line data preparation, refer to the HLMC Operator's Manual [5].

LIST OF REFERENCES

1. Barbera, A. J., Fitzgerald, M. L., Albus, J. S., "Concepts for Real-Time Sensory-Interactive Control System Architecture," Proceedings of the 14th Southeastern Symposium On Systems Theories, April, 1982.

2. Barkmeyer, E., "AMRF Standard Communications Interfaces", unpublished report, NBS, June, 1983.

3. Bunch, R., Fishman, D., The Material Buffering System of the Horizontal Workstation, to be published as an NBSIR, 1988.

4. Burton, R., Donmez, A., Scott, H., The Automated Fixturing System of the Horizontal Workstation, to be published as an NBSIR, 1988.

5. Fishman, D., The High Level Machine-Tool Controller Operator's Manual, to be published as an NBSIR, 1988.

6. Furlani et al., The Integrated Manufacturing Data Administration System (IMDAS), to be published as an NBSIR, 1988.

7. Knuth, D. E., "Fundamental Algorithms," The Art of Computer Programming, Vol. 1, pp. 442-445, Addison-Wesley, February, 1975.

8. Libes, D., Barkmeyer, E., "The Integrated Manufacturing Data Administration System (IMDAS) - an overview," International Journal of Computer Integrated Manufacturing, Vol. 1, No. 1, pp. 44-49.

9. Ragsdale, W. F., "Mass Storage Allocation In Forth," Proceedings: 1980 FORML Conference, November, 1980.

10. Rippey, W. G., "FORTH and Automation Research at the National Bureau of Standards," The Journal of Forth Application and Research, September, 1983.

11. Rybczynski, F., AMRF Network Communications, to be published as an NBSIR, 1988.

12. Scott, H., Architecture and Principles of the Horizontal Workstation, to be published as an NBSIR, 1988.

LIST OF REFERENCES (Continued)

13. Scott, H., Strouse, K., "Workstation Control in a Computer Integrated Manufacturing System," Proceedings: Autofact 6 Conference, Anaheim, CA, October, 1984.

14. Strouse, K., Burton, R., Horizontal Workstation Controller Implementation, to be published as an NBSIR, 1988.

15. Simpson, J. A., Hocken, R. J., Albus, J. S., "The Automated Manufacturing Research Facility of the National Bureau of Standards," Journal of Manufacturing Systems, Vol. 1, No. 1, 1982.

16. Wavering, A. J., Fiala, J. C., The Real-Time Control System of the Horizontal Workstation Robot, NBSIR 88-3692.

114A (REV. 2-8C)

U.S. DEPT. OF COMM. BIBLIOGRAPHIC DATA SHEET (See instructions)	1. PUBLICATION OR REPORT NO. NBSIR 88-3836	2. Performing Organ. Report No.	3. Publication Date AUGUST 1988

TITLE AND SUBTITLE

The High Level Machine-Tool Control System

AUTHOR(S)

Douglas Fishman

PERFORMING ORGANIZATION (If joint or other than NBS, see instructions)

NATIONAL BUREAU OF STANDARDS
DEPARTMENT OF COMMERCE
WASHINGTON, D.C. 20234

7. Contract/Grant No.

8. Type of Report & Period Covered

SPONSORING ORGANIZATION NAME AND COMPLETE ADDRESS (Street, City, State, ZIP)

SUPPLEMENTARY NOTES

☐ Document describes a computer program; SF-185, FIPS Software Summary, is attached.

ABSTRACT (A 200-word or less factual summary of most significant information. If document includes a significant bibliography or literature survey, mention it here)

is manual describes the High Level Machine-tool Controller (HLMC) in the Horizontal rkstation (HWS). Though the current implementation of the HLMC was designed for e HWS, there is sufficient modularity and flexibility in the system so that it may erate as a standalone automated equipment controller. In addition, this modularity ables a programmer to reprogram the HLMC so that it could operate as part of a rkstation in a factory other than the AMRF.

e manual is divided into three sections. The overview describes the physical nfiguration of the HLMC in an automated workstation, the theoretical principles hered to in integrating this component into an automated workstation, and the HLMC rdware. The architecture section is a more detailed description of all of the HLMC rdware and software interfaces. The implementation section gives the specific tails of these interfaces, pointing out relevant hardware and software modules.

is manual was written for those who are interested in the details of the design and plementation of the HLMC in the Horizontal Workstation

KEY WORDS (Six to twelve entries; alphabetical order; capitalize only proper names; and separate key words by semicolons)

tomated equipment controller, automated machining workstation, Automated Manufacturing search Facility (AMRF), Horizontal Workstation, real-time control

AVAILABILITY

☒ Unlimited

☐ For Official Distribution. Do Not Release to NTIS

☐ Order From Superintendent of Documents, U.S. Government Printing Office, Washington, D.C. 20402.

☒ Order From National Technical Information Service (NTIS), Springfield, VA. 22161

14. NO. OF PRINTED PAGES

43

15. Price

$11.95